ELSEVIER

Theoretical Computer Science 250 (2001) 219–233

Theoretical Computer Science

www.elsevier.com/locate/tcs

n and similar papers at core.ac.uk

provid

Z. Fülöp[a,*], S. Vágvölgyi[b]

[a] *Department of Computer Science, József Attila University, H-6701 Szeged, P.O. Box 652, Hungary*
[b] *Department of Applied Informatics, József Attila University, H-6701 Szeged, P.O. Box 652, Hungary*

Received August 1998; revised February 1999
Communicated by M. Ito

Abstract

We consider restricted versions of ground tree transducers: total, deterministic, and symmetric subclasses and all other subclasses created by applying any combination of these restrictions. We present the inclusion diagram of the tree transformation classes induced by these restricted ground tree transducers. We show that the following four classes of term relations are the same: (i) tree transformations induced by symmetric deterministic ground tree transducers, (ii) congruence relations on term algebras induced by reduced ground term rewriting systems, (iii) congruence relations on term algebras induced by convergent ground term rewriting systems, and (iv) finitely generated congruence relations on term algebras. As a by-product of our results, we obtain a new ground completion algorithm. Moreover, we show that the following three classes of term relations on term algebras with at least one non-nullary function symbol are also the same: (i) tree transformations induced by total symmetric deterministic ground tree transducers, (ii) congruence relations on term algebras of finite index, (iii) finitely generated congruence relations on term algebras of which the trunk is the whole set of terms. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Ground tree transducers; Ground term rewriting systems; Tree automata

1. Introduction

Dauchet et al. [4, 5] have introduced the notion of the ground tree transducer as a pair (A, B) of tree automata. The importance of ground tree transducers is in that they can simulate ground term rewriting: in [5] it was shown that for each ground term rewriting system R over a ranked alphabet Σ, one can effectively construct a ground

☆ This research was supported by the grants OTKA T 022994 of the Hungarian Research Foundation and FKFP 0095/1997 of the Hungarian Cultural and Educational Ministry.

* Corresponding author.
E-mail addresses: fulop@inf.u-szeged.hu (Z. Fülöp), vagvolgy@inf.u-szeged.hu (S. Vágvölgyi).

0304-3975/01/$ - see front matter © 2001 Elsevier Science B.V. All rights reserved.
PII: S0304-3975(99)00135-8

tree transducer (A,B) over Σ such that $\rightarrow_R^*$ is equal to the tree transformation $\tau_{(A,B)}$ induced by (A,B). Dauchet et al. [5] have used this result to show that the confluence property of ground term rewrite systems is decidable, cf. [14]. Later Dauchet and Tison showed that the first-order theory of ground term rewrite systems is also decidable [6].

In this paper we consider restricted versions of ground tree transducers. Our motivation is that studying restricted versions of a class of machines frequently gives a deeper insight into the working of the unrestricted class.

As usual for a class of tree transducers, we consider the total and the deterministic subclasses. We call a ground tree transducer (A,B) deterministic (total) if the tree automata A and B are deterministic (total). We also consider symmetric ground tree transducers, which are of the form (A,A). Moreover, we consider the eight classes of ground tree transducers obtained by combining these three properties in all possible ways.

As the first result, we compare the expressive power of the eight classes by presenting the full inclusion diagram of the tree transformation classes induced by them.

Then we show that the following four classes of term relations are the same: (i) the tree transformations induced by symmetric deterministic ground tree transducers, (ii) the congruence relations on term algebras induced by reduced ground term rewriting systems, (iii) the congruence relations on term algebras induced by convergent ground term rewriting systems, and (iv) the finitely generated congruence relations on term algebras.

In [9], Fülöp and Vágvölgyi showed the following. For every ground term equation system E over a ranked alphabet Σ, which is just a finite binary relation on T_Σ, one can effectively construct a deterministic tree automaton A over Σ such that $\leftrightarrow_E^*$, i.e., the congruence relation induced by E on the Σ-term algebra, is equal to the tree transformation $\tau_{(A,A)}$ induced by the symmetric deterministic ground tree transducer (A,A).

In the proof of the inclusion (i) $\subseteq$ (ii) we construct, for a given deterministic tree automaton A over a ranked alphabet Σ, a reduced ground term rewriting system R over Σ such that the congruence relation $\leftrightarrow_R^*$ generated by R is equal to the tree transformation $\tau_{(A,A)}$.

Thus, as a by-product of our results, we obtain a new ground completion algorithm. Given a ground term equation system E, we construct a reduced ground term rewriting system equivalent to E in two steps. In the first step, presented in [9], we compute a symmetric deterministic ground tree transducer (A,A) such that $\tau_{(A,A)} = \leftrightarrow_E^*$. Then we construct as in the proof of (i) $\subseteq$ (ii) a reduced ground term rewriting system R such that $\tau_{(A,A)} = \leftrightarrow_R^*$. Hence $\leftrightarrow_E^* = \leftrightarrow_R^*$. This ground completion parallels to Snyder's fast algorithm, see [17] and the results of the papers [15, 12].

It can easily be proved that a binary relation τ over T_Σ can be induced by a total symmetric deterministic ground tree transducer if and only if τ is a congruence of finite index on the Σ-term algebra. In addition, we show that any finitely generated congruence relation τ over the Σ-term algebra is of finite index if $trunk(\tau) = T_\Sigma$. The concept of the trunk was introduced in [10] and proved to be useful in studying congruence relations over terms. Furthermore, we prove that for a term algebra with at least one non-nullary function symbol the congruence relations of finite index are exactly the finitely generated congruence relations with trunks equal to the set of all terms.

2. Preliminaries

In this section we present a brief review of the notions, notation and preliminary results used in the paper.

Relations: A relation over a set A is a subset $\rightarrow$ of $A \times A$. We write $a \rightarrow b$ for $(a,b) \in \rightarrow$. We denote by $\rightarrow^*$ the reflexive, transitive closure and by $\leftrightarrow^*$, the reflexive, symmetric, and transitive closure of $\rightarrow$. Note that $\leftrightarrow^*$ is an equivalence relation.

A relation $\rightarrow$ is called

- Noetherian if there exists no infinite sequence of elements $a_1, a_2, a_3, \ldots$ in A such that $a_1 \rightarrow a_2 \rightarrow a_3 \rightarrow \ldots$,
- confluent if for any elements a_1, a_2, a_3 in A, whenever $a_1 \rightarrow^* a_2$ and $a_1 \rightarrow^* a_3$, there exists an element a_4 in A such that $a_2 \rightarrow^* a_4$ and $a_3 \rightarrow^* a_4$,
- Church-Rosser if, for all $a_1, a_2 \in A$, whenever $a_1 \leftrightarrow^* a_2$, there is an $a_3 \in A$ such that $a_1 \rightarrow^* a_3$ and $a_2 \rightarrow^* a_3$.
- convergent if it is Noetherian and confluent.

We shall need the following fact (cf. [2] for example).

Proposition 2.1. *A relation $\rightarrow$ is confluent if and only if it is Church–Rosser.*

Let $\rightarrow$ be a relation over a set A. An element $a \in A$ is irreducible with respect to $\rightarrow$ if there exists no $b \in A$ such that $a \rightarrow b$. It is well-known that for any convergent relation $\rightarrow$ and any class C of $\leftrightarrow^*$, C contains exactly one irreducible element a, and that for any element b in the class C, $b \rightarrow^* a$. We call a the $\rightarrow$-normal form of b.

Let ρ be an equivalence relation on A. Then for every $a \in A$, we denote by $[a]_\rho$ the ρ-class containing a, i.e. $[a]_\rho = \{ b \mid a \rho b \}$. We say that ρ is *of finite index* if the set $\{ [a]_\rho \mid a \in A \}$ is finite. Let H be a set of ρ-classes, then by $\bigcup H$ we mean $\bigcup (C \mid C \in H)$.

Terms: A ranked alphabet Σ is a finite set of symbols in which every element has a unique rank in the set of nonnegative integers. For each integer $m \geqslant 0$, Σ_m denotes the elements of Σ which have rank m. We assume that $\Sigma_0 \neq \emptyset$.

We need a countably infinite set $X = \{ x_1, x_2, \ldots \}$ of variable symbols kept fixed throughout the paper. The set of the first n elements $x_1, \ldots, x_n$ of X is denoted by X_n.

For each $n \geqslant 0$, we denote by $T_{\Sigma,n}$ the set of terms over Σ indexed by X_n. It is the smallest set U for which

(i) $\Sigma_0 \cup X_n \subseteq U$ and
(ii) $f(t_1, \ldots, t_m) \in U$ whenever $f \in \Sigma_m$ with $m \geqslant 0$ and $t_1, \ldots, t_m \in U$.

Terms are also called trees. The set $T_{\Sigma,0}$ is written simply as T_Σ and called the set of ground trees over Σ. We distinguish a subset $\tilde{T}_{\Sigma,n}$ of $T_{\Sigma,n}$ as follows: a tree $t \in T_{\Sigma,n}$ is in $\tilde{T}_{\Sigma,n}$ if and only if each variable symbol of X_n appears exactly once in t.

The tree substitution operation is defined in the following way. Given a tree $t \in T_{\Sigma,n}$ $(n \geqslant 0)$ and trees $t_1,\ldots,t_n$, we denote by $t[t_1,\ldots,t_n]$ the tree which can be obtained from t by replacing each occurrence of x_i in t by t_i, for $1 \leqslant i \leqslant n$.

For a ground term $t \in T_\Sigma$, the set $sub(t)$ of subtrees of t is defined by recursion as follows:

(i) if $t \in \Sigma_0$, then $sub(t) = \{t\}$,

(ii) if $t = f(t_1,\ldots,t_m)$ for some $m \geqslant 1$, $f \in \Sigma_m$, and $t_1,\ldots,t_m \in T_\Sigma$, then we have $sub(t) = \bigcup(sub(t_i) \mid 1 \leqslant i \leqslant m) \cup \{t\}$.

For a tree language $L \subseteq T_\Sigma$, the set $sub(L)$ of subtrees of elements of L is defined by the equation $sub(L) = \bigcup(sub(t) \mid t \in L)$. We say that L is *closed under subtrees* if $sub(L) \subseteq L$.

Algebras: Let Σ be a ranked alphabet. A Σ algebra is a system $\mathbf{B} = (B, \Sigma^{\mathbf{B}})$, where B is a nonempty set, called the carrier set of $\mathbf{B}$, and $\Sigma^{\mathbf{B}} = \{f^{\mathbf{B}} \mid f \in \Sigma\}$ is a Σ-indexed set of operations over B such that for every $f \in \Sigma_m$ with $m \geqslant 0$, $f^{\mathbf{B}}$ is a mapping from B^m to B.

An equivalence relation $\rho \subseteq B \times B$ is a congruence on $\mathbf{B}$ if $f^{\mathbf{B}}(t_1,\ldots,t_m)\rho f^{\mathbf{B}}(p_1,\ldots,p_m)$ whenever $f \in \Sigma_m$, $m \geqslant 0$, and $t_i \rho p_i$, for $1 \leqslant i \leqslant m$. The least congruence on $\mathbf{B}$ containing a given relation $\sigma \subseteq B \times B$ is called the *congruence generated by* σ. A congruence on $\mathbf{B}$ is finitely generated if it is generated by a finite relation $\sigma \subseteq B \times B$.

In this paper we shall mainly deal with the algebra $\mathbf{TA} = (T_\Sigma, \Sigma)$ of terms over Σ, where for $f \in \Sigma_m$ with $m \geqslant 0$ and $t_1,\ldots,t_m \in T_\Sigma$, we have $f^{\mathbf{TA}}(t_1,\ldots,t_m) = f(t_1,\ldots,t_m)$.

We adopt the concepts of a simple class and of a compound class of a congruence ρ on the term algebra $\mathbf{TA}$ from Fülöp and Vágvölgyi [10]. Informally, these concepts are defined as follows. Clearly, every ρ-class C can be written as the union of sets of the form $f(C_1,\ldots,C_m)$ for some suitable f's and classes $C_1,\ldots,C_m$. Especially, if the union has only one member, i.e., $C = f(C_1,\ldots,C_m)$, then C is called a simple class. If a class is not simple, then it is compound.

More formally, given a congruence ρ on $\mathbf{TA}$, a ρ-class C is called *simple* if for any function symbols $f \in \Sigma_m, g \in \Sigma_n$, with $m,n \geqslant 0$ and ρ-classes $C_1,\ldots,C_m,C'_1,\ldots,C'_n$, if $f^{\mathbf{TA}}(C_1,\ldots,C_m) \subseteq C$ and $g^{\mathbf{TA}}(C'_1,\ldots,C'_n) \subseteq C$, then $f = g$, $m = n$, $C_1 = C'_1,\ldots,C_m = C'_m$. If a ρ-class C is not simple then it is called a *compound* class. The set of all compound classes is denoted by $comp(\rho)$.

Next we adopt the trunk of a congruence ρ from Fülöp and Vágvölgyi [10]. Let ρ be a congruence on $\mathbf{TA}$, the trunk $trunk(\rho)$ of ρ is the set $sub(\bigcup comp(\rho))$.

Ground term rewriting systems and equation systems: A ground term rewriting system (gtrs) over a ranked alphabet Σ is a finite subset R of $T_\Sigma \times T_\Sigma$. The elements of R, called rules, can be used to define a relation, called rewriting relation, $\rightarrow_R$ introduced as follows: for any $p, q \in T_\Sigma$, we have $p \rightarrow_R q$ if and only if there exists a rule, (u, v) in R and a context $c \in \tilde{T}_{\Sigma,1}$ such that $p = c[u]$ and $q = c[v]$. The rules in R will be written in the form $u \rightarrow v$ as well. Moreover, we say that u is the left-hand side and v is the right-hand side of the rule $u \to v$. Besides the "one-way" relations $\to_R$ and $\to_R^*$ we also consider the congruence relation on **TA** generated by R, which is $\leftrightarrow_R^*$.

We say that R is Noetherian, (confluent, etc.) if $\to_R$ is Noetherian (confluent, etc.). A term $t \in T_\Sigma$ is irreducible with respect to R if it is irreducible with respect to $\to_R$. A gtrs R is reduced if for every rule $u \to v$ in R, u is irreducible with respect to $R - \{u \to v\}$ and v is irreducible with respect to R.

We recall the following two important results.

Proposition 2.2 (Snyder [17]). *Any reduced gtrs R is convergent.*

A ground term equation system E over a ranked alphabet Σ is also a finite binary relation on T_Σ. However, in case of a ground term equation system E we consider only the congruence relation on **TA** generated by E, which is denoted by $\leftrightarrow_E^*$. We say that a gtrs R over Σ is equivalent to E if $\leftrightarrow_R^* = \leftrightarrow_E^*$.

Proposition 2.3 (Snyder [17]). *For a ground term equation system E over a ranked alphabet Σ one can effectively construct an equivalent reduced gtrs R over Σ.*

For a gtrs R over a ranked alphabet Σ, by the set of subterms occurring in R we mean the set

$$sub(R) = \bigcup\{sub(u) \cup sub(v) \mid u \to v \text{ is in } R\}.$$

We now recall a result on the trunk of a congruence generated by a reduced gtrs.

Proposition 2.4 (Vágvölgyi [20]). *Let R be a reduced gtrs over a ranked alphabet Σ. Then*

$$trunk(\underset{R}{\overset{*}{\leftrightarrow}}) = \bigcup\{\, [t]_{\leftrightarrow_R^*} \mid t \in sub(R)\}.$$

Tree automata: Let Σ be a ranked alphabet. A tree automaton A over Σ is a gtrs over the ranked alphabet $\Sigma \cup STATES_A$, where $STATES_A$, the state set of A, consists of nullary function symbols, $STATES_A \neq \emptyset$ and $STATES_A \cap \Sigma = \emptyset$. Moreover, each rule in A is of the form

$$f(a_1, \ldots, a_n) \to a \qquad \text{(called a reduction rule)},$$

where $f \in \Sigma_n$, $n \geqslant 0$, $a, a_1, \ldots, a_n \in STATES_A$, or is of the form

$$a_1 \to a_2 \qquad \text{(called a } \lambda\text{-rule)},$$

where $a_1, a_2 \in STATES_A$.

A state $a \in STATES_A$ is reachable if there is a tree $t \in T_\Sigma$ such that $t \to_A^* a$. The following can be shown by applying well-known techniques of tree automaton theory, see [11].

Proposition 2.5. *Let $a \in STATES_A$. It is decidable if the state a is reachable. Moreover, if a is reachable, then one can effectively construct a tree $s \in T_\Sigma$ such that $s \to_A^* a$.*

We say that a tree automaton A over Σ is connected if each state in $STATES_A$ is reachable. Moreover, A is deterministic if for any $f \in \Sigma_m$, $m \geqslant 0$, $a_1, \ldots, a_m \in STATES_A$, there is at most one rule with left-hand side $f(a_1, \ldots, a_m)$ in A, and there are no λ-rules in A. Finally, A is total if for any $f \in \Sigma_m$, $m \geqslant 0$, $a_1, \ldots, a_m \in STATES_A$, there is at least one rule with left-hand side $f(a_1, \ldots, a_m)$ in A.

Proposition 2.6. *Let Σ be a ranked alphabet, and let A be a deterministic tree automaton over Σ. Then A is a reduced gtrs over the ranked alphabet $\Sigma \cup STATES_A$.*

Proof. By direct inspection of the rules of A. $\square$

For results on the connection between gtrs's and tree automata, see [1, 3, 7–10, 13, 18–21].

Ground tree transducers: Ground tree transducers [5, 6] proved to be an efficient tool in the theory of ground term rewriting systems. A ground tree transducer (gtt) over Σ is a pair (A, B) of tree automata over Σ. The tree transformation $\tau_{(A,B)} \subseteq T_\Sigma \times T_\Sigma$ induced by (A, B) is defined as follows. For any trees $p, q \in T_\Sigma$, $(p, q) \in \tau_{(A,B)}$ if and only if there exist a tree $u \in \tilde{T}_\Sigma(X_n)$, $n \geqslant 0$, and trees $z_1, \ldots, z_n, z'_1, \ldots, z'_n \in T_\Sigma$ and common states $a_1, \ldots, a_n$ of A and B such that

$$p = u[z_1, \ldots, z_n] \xrightarrow[A]{*} u[a_1, \ldots, a_n] \quad \text{and} \quad q = u[z'_1, \ldots, z'_n] \xrightarrow[B]{*} u[a_1, \ldots, a_n],$$

where $z_i \to_A^* a_i$ and $z'_i \to_A^* a_i$ for $1 \leqslant i \leqslant n$.

Dauchet et al. [5] have shown that for each gtrs R over Σ, one can effectively construct a ground tree transducer (A, B) over Σ such that $\to_R^* = \tau_{(A,B)}$.

3. The results

We introduce the following restrictions for ground tree transducers.

Definition 3.1. A ground tree transducer (A, B) is called

(i) deterministic, if A and B are deterministic,
(ii) total, if A and B are total, and
(ii) symmetric, if $A = B$.

Any combination of the restrictions can also be formed. For example, a ground tree transducer is symmetric deterministic if it is both symmetric and deterministic. Thus together with the unrestricted class, we have defined eight tree transformation classes.

Table 1

	Reflexive	Symmetric	Transitive
GTT, *T-GTT*, *D-GTT*, and *TD-GTT*	+	−	−
S-GTT and *TS-GTT*	+	+	−
SD-GTT and *TSD-GTT*	+	+	+

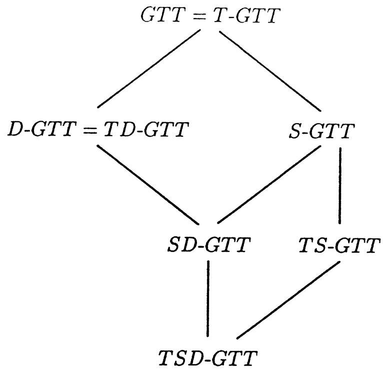

Fig. 1. The inclusion diagram of *GTT*, *D-GTT*, *S-GTT*, *SD-GTT* and *TSD-GTT*.

We denote by *GTT* the class of tree transformations induced by ground tree transducers. The prefixes *D*-, *S*-, and *T*- denote the subclasses of *GTT* induced by deterministic, symmetric and total ground tree transducers, respectively. Clearly, any combination of the prefixes can be formed and for example *SD-GTT* stands for the class of tree transformations induced by symmetric deterministic ground tree transducers.

We are going to present the inclusion diagram of the eight tree transformation classes induced by the ground tree transducers introduced above. In order to verify these inclusions it is useful to know about the basic properties reflexivity, symmetry, and transitivity of the various tree transformation classes. Table 1 displays this information. The table is organized such that if all tree transformations in the class Y-*GTT*, where $Y \in \{\ \ , T, D, TD, S, TS, SD, TSD\}$, have property $\mathscr{P} \in \{\text{reflexive}, \text{symmetric}, \text{transitive}\}$, then the entry determined by Y-*GTT* and $\mathscr{P}$ contains the sign $+$, otherwise the entry contains the sign $-$. We can easily verify the correctness of Table 1.

Theorem 3.2. *The inclusion diagram of the tree transformation classes GTT, T-GTT, D-GTT, TD-GTT, S-GTT, TS-GTT, SD-GTT and TSD-GTT is the one in Fig.* 1.

Proof. The equalities of $GTT = T\text{-}GTT$ and $D\text{-}GTT = TD\text{-}GTT$ can be proved by applying a standard construction: given a gtt (A, B) we construct a total gtt (A', B') such that $\tau_{(A,B)} = \tau_{(A',B')}$ as follows. We add new states a and b with $a \neq b$ to $STATES_A$ and $STATES_B$, respectively. Moreover, we add new rules to A and B with right-hand sides a and b, respectively and with all possible left-hand sides for which there are no rules in A and B. The construction preserves determinism.

The other inclusions shown by the diagram hold by definition. In order to prove that they are proper (strict) and that the unrelated classes are incomparable it is sufficient to show that the classes *SD-GTT* and *TS-GTT* as well as the classes *D-GTT* and *S-GTT* are incomparable.

First we prove that *SD-GTT* and *TS-GTT* are incomparable by showing that *TS-GTT* $-$ *SD-GTT* $\neq \emptyset$ and *SD-GTT* $-$ *TS-GTT* $\neq \emptyset$. The first inequality should be clear because all relations in *SD-GTT* are transitive and there are relations in *TS-GTT* which are not, cf. Table 1.

Next we give a deterministic tree automaton A such that the symmetric deterministic ground tree transformation $\tau_{(A,A)}$ is not in *TS-GTT*. Hence *SD-GTT* $-$ *TS-GTT* $\neq \emptyset$ follows. So let Σ be a ranked alphabet with at least one non-nullary symbol and let A be the tree automaton with $A=\emptyset$. Hence A has no rules and thus $\tau_{(A,A)}=\{(t,t)\mid t\in T_\Sigma\}$. Now assume, by contradiction, that $\tau_{(A,A)}\in$ *TS-GTT*, i.e., that there is a total tree automaton B such that $\tau_{(A,A)}=\tau_{(B,B)}$, where (B,B) is the total symmetric ground tree transducer created from B. Since B is total and has a finite number of states, there are different trees $p,q\in T_\Sigma$, a state $b\in STATES_B$ such that $p\rightarrow_B^* b$ and $q\rightarrow_B^* b$. Hence $(p,q)\in\tau_{(B,B)}$, which is a contradiction because $(p,q)\notin\tau_{(A,A)}$.

In order to show that the classes *D-GTT* and *S-GTT* are incomparable, it is sufficient to show that *D-GTT* $-$ *S-GTT* $\neq\emptyset$ and *S-GTT* $-$ *D-GTT* $\neq\emptyset$. The first inequality obviously holds, because certainly there are deterministic ground tree transformations which are not symmetric.

We prove the second one by giving a symmetric ground tree transducer (A,A) such that $\tau_{(A,A)}\notin$ *D-GTT*. Let $\Sigma=\Sigma_0\cup\Sigma_1$, where $\Sigma_0=\{\#,\$\}$ and $\Sigma_1=\{f\}$, and let $STATES_A=\{a,b\}$. Further, let A be a tree automaton over Σ having the rules $\#\rightarrow a$, $f(a)\rightarrow b$, and $\$\rightarrow a\mid b$. Note that A is nondeterministic. Then $(\#,\$),(f(\#),\$)\in\tau_{(A,A)}$. Now let us assume that $\tau_{(A,A)}\in$ *D-GTT*, i.e., that $\tau_{(A,A)}=\tau_{(B,C)}$ for some deterministic tree automata B and C. Since $(\#,\$)\in\tau_{(B,C)}$, there is a common state c of B and C such that $\#\rightarrow c\in B$ and $\$\rightarrow c\in C$. Moreover, since $(f(\#),\$)\in\tau_{(B,C)}$ the rule $f(c)\rightarrow c$ must be also in B. Then, for every $n\geqslant 1$, $(f^n(\#),\$)\in\tau_{(B,C)}$, a contradiction because this does not hold for $\tau_{(A,A)}$. $\square$

Next we prove our main result.

Theorem 3.3. *The following four statements are equivalent for any binary relation τ on T_Σ.*

(a) $\tau\in$ *SD-GTT*,
(b) τ *is a congruence on* **TA** *generated by a reduced gtrs*,
(c) τ *is a congruence on* **TA** *generated by a convergent gtrs*,
(d) τ *is a finitely generated congruence on* **TA**.

Proof. First step: (a) implies (b). Let A be a deterministic tree automaton. We construct a reduced gtrs R such that $\tau_{(A,A)}=\leftrightarrow_R^*$.

Without loss of generality, we can assume that A is connected. First, for every state a of A, we construct a tree $tree(a) \in T_\Sigma$, such that $tree(a) \rightarrow_A^* a$. The trees $tree(a)$ are constructed by the following algorithm. $\square$

Algorithm 3.4

Input: A connected deterministic tree automaton A.

Output: For every $a \in STATES_A$, a tree $tree(a) \in T_\Sigma$ such that $tree(a) \rightarrow_A^* a$.

Auxiliary variables: For every $a \in STATES_A$, a boolean flag $flag(a)$, which indicates whether or not $tree(a)$ has been defined.

for every $a \in STATES_A$, let $flag(a) = false$;
while there is an $a \in STATES_A$ with $flag(a) = false$ **do**
for every rule $f(a_1, \ldots, a_m) \rightarrow a$ in R **do**
if $flag(a_1) = \cdots = flag(a_m) = true$ and $flag(a) = false$,
  then begin
    $tree(a) := f(tree(a_1), \ldots, tree(a_m))$;
    $flag(a) := true$
  end

As A is connected, Algorithm 3.4 terminates with $flag(a) = true$, for all $a \in STATES_A$.

Next we give R. Let

$$R = \{f(tree(a_1), \ldots, tree(a_m)) \rightarrow tree(a) \mid f(a_1, \ldots, a_m) \rightarrow a \in A \text{ and } f(tree(a_1), \ldots, tree(a_m)) \neq tree(a)\}.$$

We state and prove properties P1–P7 of the trees $tree(a)$ and R, of which P5 states that R is reduced and P7 states that $\tau_{(A,A)} = \leftrightarrow_R^*$.

P1: For every $a \in STATES_A$, $tree(a) \rightarrow_A^* a$. This can be seen by direct inspection of Algorithm 304.

P2: For every $u \rightarrow v \in R$, the relation $u \leftrightarrow_A^* v$ holds. This follows from P1, and the fact that all rules in R have the form $f(tree(a_1), \ldots, tree(a_m)) \rightarrow tree(a)$, where $f(a_1, \ldots, a_m) \rightarrow a \in A$.

P3: For all $a, b \in STATES_A$, $a = b$ if and only if $tree(a) = tree(b)$. This can be seen as follows. If $a = b$ then certainly $tree(a) = tree(b)$ because, for every a, Algorithm 3.4 constructs exactly one tree $tree(a)$. To show the converse, let us assume $a \neq b$ and $tree(a) = tree(b)$. Then, by $tree(a) \rightarrow_A^* a$ and $tree(b) \rightarrow_A^* b$, two different states, viz. a and b, can be derived with rules in A from the same tree $tree(a) = tree(b)$ contradicting that A is deterministic.

P4: For every $a \in STATES_A$, the tree $tree(a)$ is irreducible with respect to R. We prove this by induction on the structure of $tree(a)$.

Base: Let $a \in STATES_A$ and $tree(a) = f$, where $f \in \Sigma_0$. Then $f \rightarrow a \in A$. Assume f is not irreducible, i.e., there is a rule $f \rightarrow v$ in R. Then, by the definition of R, there is a rule $f \to b$ in A such that $v = tree(b)$, $f \neq v$. Then, $a = b$, because A is deterministic and thus $f = tree(a) = tree(b) = v$. This contradicts the definition of R.

Induction step: Let $a \in STATES_A$ and $tree(a) = f(t_1, \dots, t_m)$, where $m \geqslant 1$, $f \in \Sigma_m$ and $t_1, \dots, t_m \in T_\Sigma$. Let us denote $f(t_1, \dots, t_m)$ by u. Then, by Algorithm 3.4, $u = f(tree(a_1), \dots, tree(a_m))$ for some rule $f(a_1, \dots, a_m) \to a$ in A. Let us assume that u is not irreducible. Then, by the induction hypothesis, the trees $tree(a_i)$ are irreducible, therefore there must be a rule $u \to v$ in R. By the definition of R, there is a rule $f(b_1, \dots, b_m) \to b$ in A such that $u = f(tree(b_1), \dots, tree(b_m))$ and $v = tree(b)$, $u \neq v$. Since $tree(a_i) = tree(b_i)$ we obtain $a_i = b_i$ for all $1 \leqslant i \leqslant m$. Then $a = b$ because A is deterministic, and thus by P3 $u = tree(a) = tree(b) = v$. A contradiction again.

P5: R is reduced. Let $u \to v$ be an arbitrary rule in R. Then there is a rule $f(a_1, \dots, a_m) \to a$ in A such that $u = f(tree(a_1), \dots, tree(a_m))$ and $v = tree(a)$. By P4, v is irreducible. Now assume that u is not irreducible with respect to $R - \{u \to v\}$. Since $u = f(tree(a_1), \dots, tree(a_m))$ and the trees $tree(a_i)$ are irreducible, there must be a rule $u \to v'$ in $R - \{u \to v\}$. Then, by the definition of R, there is a rule $f(b_1, \dots, b_m) \to b$ in A such that $u = f(tree(b_1), \dots, tree(b_m))$ and $v' = tree(b)$. Moreover, $a_i \neq b_i$, for some $1 \leqslant i \leqslant m$. (Otherwise $a = b$ and $v = v'$ because A is deterministic.) Then, by P3, $tree(a_i) \neq tree(b_i)$, hence $u = f(tree(a_1), \dots, tree(a_m)) \neq f(tree(b_1), \dots, tree(b_m)) = u$, a contradiction.

P6: For all $p \in T_\Sigma$ and $a \in STATES_A$, $p \to_A^* a$ if and only if $p \to_R^* tree(a)$.

First we prove the "only if" clause by induction on p.

Base: Let $p = f$ for some $f \in \Sigma_0$ and assume $f \to_A^* a$, i.e., that $f \to a \in A$. Then either $f = tree(a)$ or there is a rule $f \to tree(a) \in R$. In both cases, $p \to_R^* tree(a)$.

Induction step: Let $p = f(p_1, \dots, p_m)$. Then, $p = f(p_1, \dots, p_m) \to_A^* f(a_1, \dots, a_m) \to_A a$. By induction hypothesis, for every $1 \leqslant i \leqslant m$, $p_i \to_R^* tree(a_i)$. Moreover, since $f(a_1, \dots, a_m) \to a \in A$, either $tree(a) = f(tree(a_1), \dots, tree(a_m))$ or $f(tree(a_1), \dots, tree(a_m)) \to tree(a) \in R$. Again, in both cases, $p \to_R^* tree(a)$.

The "if" clause can be proved also by induction.

Base: Let $p = f$ for some $f \in \Sigma_0$ and assume $f \to_R^* tree(a)$. Then either $f = tree(a)$ or $f \to tree(a) \in R$. In the first case $f \to_A^* a$, by P1, in the second case $f \to a \in A$, hence again $f \to_A^* a$.

Induction step: Let $p = f(p_1, \dots, p_m)$ and assume $p \to_R^* tree(a)$. Subcase (a): let us assume that the root symbol of p is also rewritten. Then $p \to_R^* tree(a)$ can be written in the form $p = f(p_1, \dots, p_m) \to_R^* f(tree(a_1), \dots, tree(a_m)) \to_R tree(a)$. By the induction hypothesis, for every $1 \leqslant i \leqslant m$, $p_i \to_A^* a_i$. On the other hand, by the definition of R, $f(a_1, \dots, a_m) \to a \in A$. Then we obtain $p = f(p_1, \dots, p_m) \to_A^* f(a_1, \dots, a_m) \to_A a$. Subcase (b): the root symbol of p is not rewritten. Then $p \to_R^* tree(a)$ can be written in the form

$$p = u[p_1, \dots, p_n] \xrightarrow[R]{*} u[tree(a_1), \dots, tree(a_n)] = tree(a),$$

where $n \geqslant 1$, $u \in \tilde{T}_\Sigma(X_n)$, and, for every $1 \leqslant i \leqslant n$, $p_i \to_R^* tree(a_i)$ such that the root of p_i is rewritten. Then as we proved in subcase (a), $p_i \to_A^* a_i$, for all $1 \leqslant i \leqslant n$ and thus $p = u[p_1, \dots, p_n] \to_A^* u[a_1, \dots, a_n]$. On the other hand, since $u[tree(a_1), \dots, tree(a_n)] =$

$tree(a)$ we obtain by P1 that $u[tree(a_1),\dots,tree(a_n)] \to_A^* a$. Moreover, also by P1, $tree(a_i) \to_A^* a_i$, for every $1 \leqslant i \leqslant n$, and thus $u[tree(a_1),\dots,tree(a_n)] \to_A^* u[a_1,\dots,a_n]$. By Propositions 2.2 and 2.6, A is convergent. As a is irreducible for A, we obtain $u[a_1,\dots,a_n] \to_A^* a$. This means that $p = u[p_1,\dots,p_n] \to_A^* u[a_1,\dots,a_n] \to_A^* a$.

P7: $\tau_{(A,A)} = \leftrightarrow_R^*$.

First we show that $\tau_{(A,A)} \subseteq \leftrightarrow_R^*$. If $(p,q) \in \tau_{(A,A)}$, then there exist trees $u \in \tilde{T}_\Sigma(X_n)$, $n \geqslant 0$ and $z_1,\dots,z_n,z_1',\dots,z_n' \in T_\Sigma$ and states $a_1,\dots,a_n$ of A such that

$$p = u[z_1,\dots,z_n] \xrightarrow[A]{*} u[a_1,\dots,a_n] \quad \text{and} \quad q = u[z_1',\dots,z_n'] \xrightarrow[A]{*} u[a_1,\dots,a_n],$$

where $z_i \to_A^* a_i$ and $z_i' \to_A^* a_i$ for $1 \leqslant i \leqslant n$. Then, by the property P6 of R, for the tree $r = u[tree(a_1),\dots,tree(a_n)] \in T_\Sigma$ the relations $p \to_R^* r$ and $q \to_R^* r$ hold proving $p \leftrightarrow_R^* q$.

Next we show that $\leftrightarrow_R^* \subseteq \tau_{(A,A)}$. Let us assume $p \leftrightarrow_R^* q$. Then $p \leftrightarrow_A^* q$ by P2. By Proposition 2.6, A is a reduced gtrs over $\Sigma \cup STATES_A$ and thus, by Propositions 2.2 and 2.1, it is Church-Rosser, hence there exists a tree $r' \in T_{\Sigma \cup STATES_A}$ such that $p \to_A^* r'$ and $q \to_A^* r'$. This proves $(p,q) \in \tau_{(A,A)}$.

Second step: (b) implies (c) by Proposition 2.2.

Third step: (c) implies (d) by definition.

Fourth step: (d) implies (a). For any finite subset σ of $T_\Sigma \times T_\Sigma$, we get by a construction obtained in [9] a deterministic tree automaton A such that $\rho = \tau_{(A,A)}$, where ρ is the congruence on **TA** generated by σ.

In fact, in [9], for a finite relation $E \subseteq T_\Sigma \times T_\Sigma$, called a ground tree equation system in that paper, we constructed a reduced gtrs R for solving the word problem of E in the following way. We constructed a finite set C disjoint with Σ. We considered elements of C as nullary symbols and constructed a gtrs R over $\Sigma \cup C$ having rules of the form $f(c_1,\dots,c_m) \to c$, where $c,c_1,\dots,c_m \in C$. Moreover, we proved that R is reduced, (Proposition 3.1 of Fülöp and Vágvölgyi [9]) and that $\leftrightarrow_E^* = \leftrightarrow_R^* \cap T_\Sigma \times T_\Sigma$, where $\leftrightarrow_E^*$ is the congruence relation on **TA** generated by E, (Theorem 3.4 of Fülöp and Vágvölgyi [9]). Hence R is also convergent by Proposition 2.2. Thus R can be used to solve the word problem of E in the following way. For any terms $p,q \in T_\Sigma$, we compute their $\to_R$-normal forms $p',q' \in T_{\Sigma \cup C}$, respectively. Then we have $p \leftrightarrow_E^* q$ if and only if $p' = q'$. This implies the following property. For any terms $p,q \in T_\Sigma$, $p \leftrightarrow_E^* q$ if and only if there is a term r over $\Sigma \cup C$ such that $p \to_R^* r$ and $q \to_R^* r$.

Now we observe that R, using the terminology of this paper, is a deterministic tree automaton over Σ where $STATES_R = C$. Moreover, the fact that R, as a gtrs, is reduced means that it, as a tree automaton, is deterministic. This is because the condition that, for every rule $f(c_1,\dots,c_m) \to c$, the left-hand side $f(c_1,\dots,c_m)$ is irreducible with respect to $R - \{f(c_1,\dots,c_m) \to c\}$ means that there are no different rules in R with the same left-hand side. Moreover, by direct inspection of the definition of R, we obtain that there are no λ-rules in R.

Finally, we observe that $\leftrightarrow_E^* = \tau_{(R,R)}$ because for any terms $p,q \in T_\Sigma$, $p \leftrightarrow_E^* q$ if and only if there is a term r over $\Sigma \cup C$ such that $p \to_R^* r$ and $q \to_R^* r$. Hence (d) implies (a). $\square$

The proof of Theorem 3.3 yields a new ground completion algorithm as follows. Given a ground term equation system E, we construct a reduced ground term rewriting system equivalent to E in two steps. In the first step, presented in [9], we compute a symmetric deterministic ground tree transducer (A,A) such that $\tau_{(A,A)} = \leftrightarrow_E^*$. Then in the second step, we construct a reduced ground term rewriting system R such that $\tau_{(A,A)} = \leftrightarrow_R^*$. Hence $\leftrightarrow_E^* = \leftrightarrow_R^*$. This ground completion parallels to Snyder's fast algorithm, see [17], and the results of [12, 15].

The following statement characterizes the total deterministic symmetric ground tree transformations in terms of congruences of finite index. It can be proved by a standard construction.

Theorem 3.5. *A binary relation τ is a congruence on* **TA** *of finite index if and only if $\tau \in$ TSD-GTT.*

The following lemma is an immediate consequence of Proposition 2.4.

Lemma 3.6. *If R is a reduced gtrs over a ranked alphabet Σ such that trunk$(\leftrightarrow_R^*) = T_\Sigma$, then $\leftrightarrow_R^*$ is of finite index.*

Theorem 3.7. *Any finitely generated congruence relation τ over the term algebra* **TA** *with trunk$(\tau) = T_\Sigma$ is of finite index.*

Proof. It easily follows from Proposition 2.3 and Lemma 3.6. $\square$

Now we give the following characterization of total symmetric deterministic ground tree transformations.

Theorem 3.8. *Let Σ be a ranked alphabet with $\Sigma - \Sigma_0 \neq \emptyset$. Then the following three statements are equivalent for any binary relation τ on T_Σ.*
(a) *$\tau \in$ TSD-GTT,*
(b) *τ is a congruence on* **TA** *of finite index,*
(c) *τ is a finitely generated congruence on* **TA** *with trunk$(\tau) = T_\Sigma$.*

Proof. The equivalence of (a) and (b) is stated by Theorem 3.5.

Next we show that (b) implies (c). Let τ be an arbitrary congruence on **TA** of finite index. By the results of Rival and Sands [16], τ is finitely generated, cf. Lemma 4.6 in [8]. By Proposition 2.3, τ is a congruence on **TA** generated by a reduced gtrs R, i.e., $\tau = \leftrightarrow_R^*$. Hence by Proposition 2.4,

$$trunk(\tau) = trunk(\overset{*}{\underset{R}{\leftrightarrow}}) = \bigcup \{ [t]_{\leftrightarrow_R^*} \mid t \in sub(R) \} = \bigcup \{ [t]_\tau \mid t \in sub(R) \}. \tag{1}$$

We now show that $trunk(\tau) = T_\Sigma$. Assume for contradiction that $T_\Sigma - trunk(\tau) \neq \emptyset$. Take a tree $t \in T_\Sigma - trunk(\tau)$ of minimal height. We distinguish two cases.

Case 1: $t = g \in \Sigma_0$. In this case we define trees $u_n \in T_\Sigma$, $n \geqslant 1$, as follows. Take any symbol $f \in \Sigma - \Sigma_0$. Let $u_1 = g$, and let $u_n = f(u_{n-1}, g, \ldots, g)$ for every $n > 1$. Observe that, for every $n \geqslant 1$, g is a subtree of u_n and since $trunk(\tau)$ is closed under subtrees, $u_n \notin trunk(\tau)$. By the definition of $trunk(\tau)$, $[u_n]_\tau$ is a simple τ-class for $n \geqslant 1$. By induction on k, we show that for each $k \geqslant 1$, for all $1 \leqslant i < j \leqslant k$, $(u_i, u_j) \notin \tau$.

Base: $k = 1$. Trivial.

Induction step: Let $k \geqslant 2$. By the induction hypothesis, it is sufficient to show that the trees u_i and u_k are not in the relation τ for $1 \leqslant i < k$. Since $u_1 = g$ and $u_k = f(u_{k-1}, g, \ldots, g)$, and $[u_k]_\tau$ is a simple τ-class, we obtain $(u_1, u_k) \notin \tau$. Let $2 \leqslant i < k$. Recall that $u_i = f(u_{i-1}, g, \ldots, g)$ and $u_k = f(u_{k-1}, g, \ldots, g)$. By the induction hypothesis, $(u_{i-1}, u_{k-1}) \notin \tau$. As $[u_k]_\tau$ is a simple τ-class, $(u_i, u_k) \notin \tau$.

Case 2: $t = f(t_1, \ldots, t_m)$, $m \geqslant 1$, $f \in \Sigma_m$, $t_1, \ldots, t_m \in T_\Sigma$. As $t \in T_\Sigma - trunk(\tau)$ is of minimal height, $t_1, \ldots, t_m \in trunk(\tau)$. Let $u_1 = t$. For each $n \geqslant 2$, let $u_n = f(u_{n-1}, t_2, \ldots, t_m)$. Observe that $u_1 = t$ is a subtree of u_n for $n \geqslant 1$. As $trunk(\tau)$ is closed under subtrees, $u_n \notin trunk(\tau)$ for $n \geqslant 1$. By induction on k, we show that for each $k \geqslant 1$, for all $1 \leqslant i < j \leqslant k$, $(u_i, u_j) \notin \tau$.

Base: $k = 1$. Trivial.

Induction step: Let $k \geqslant 2$. By the induction hypothesis, it is sufficient to show that the trees u_i and u_k are not in the relation τ for $1 \leqslant i < k$. Now $u_1 = f(t_1, t_2, \ldots, t_m)$ and $u_k = f(u_{k-1}, t_2, \ldots, t_m)$. As $t_1 \in trunk(\tau)$ and $u_k \notin trunk(\tau)$, by the equality (1), $(u_1, u_k) \notin \tau$. Let $2 \leqslant i < k$. Since $u_i = f(u_{i-1}, t_2, \ldots, t_m)$ and $u_k = f(u_{k-1}, t_2, \ldots, t_m)$, by the induction hypothesis, $(u_{i-1}, u_{k-1}) \notin \tau$. Since $[u_k]_\tau$ is a simple class, $(u_i, u_k) \notin \tau$.

In both cases we obtained that τ is not of finite index, a contradiction. Thus Condition (b) holds.

The implication (c) $\Rightarrow$ (b) was already proved in Theorem 3.7. □

In case $\Sigma = \Sigma_0$ condition (c) implies (b) because T_Σ is finite and hence any congruence on **TA** is of finite index. However, the converse does not hold: let $\Sigma = \Sigma_0 = \{\#, \$\}$ and let $\tau = \{(\#, \#), (\$, \$)\}$. Certainly, τ is a congruence on **TA**, it is generated by $\emptyset$ but $trunk(\tau) = \emptyset \neq T_\Sigma$.

Finally, we mention that Theorem 3.8 gives a characterization of congruence relations of finite index of term algebras in terms of their trunk. In fact, it follows from the results of Rival and Sands [16] that a congruence relation of finite index of a term algebra is also finitely generated. It is not hard to show by a counter-example that the converse is not true. However, we obtain the following characterization of congruences of finite index.

Theorem 3.9. *Let Σ be a ranked alphabet with $\Sigma - \Sigma_0 \neq \emptyset$. A finitely generated congruence relation τ on* **TA** *is of finite index, if and only if $trunk(\tau) = T_\Sigma$.*

Proof. Follows from the equivalence of statements (b) and (c) of Theorem 3.8. □

4. Conclusion

In this paper we considered restricted versions of ground tree transducers. Altogether eight ground tree transducer classes were considered: the unrestricted one and the ones obtained by applying any combination of the restrictions deterministic, total, and symmetric. We established the inclusion diagram of the tree transformation classes induced by the eight ground tree transducer classes. We showed that symmetric deterministic ground tree transformations are the same as finitely generated congruences of term algebras. As a by-product, we obtained a new ground completion algorithm. We also showed that congruences of term algebras of finite index are the same as tree transformations induced by total deterministic symmetric ground tree transducers.

Acknowledgements

The authors are indebted to the unknown referees for their valuable suggestions to improve the submitted version of this paper.

References

[1] J. Apró, S. Vágvölgyi, The Ground Tree Transducer Game with Identical Tree Automata, submitted for publication.
[2] R. V. Book, F. Otto, String-Rewriting Systems, Texts and Monographs in Computer Science, Springer Publishing Company, Berlin, 1993.
[3] W.S. Brainerd, Tree generating regular systems, Inform. and Control 14 (1969) 217–231.
[4] H. Comon, M. Dauchet, R. Gilleron, D. Lugiez, S. Tison, M. Tommasi, Tree Automata Techniques and Applications, work in progress, http://l3ux02.univ-lille3.fr/tata.
[5] M. Dauchet, P. Heuillard, P. Lescanne, S.Tison, Decidability of the confluence of finite ground term rewrite systems and of other related term rewrite systems, Inform. and Control 88 (1990) 187–201.
[6] M. Dauchet, S. Tison, The Theory of Ground Rewrite Systems is Decidable, Proc. of the 5th IEEE Symp. on Logic in Computer Science, Philadelphia, 1990, pp. 242–248.
[7] J. Engelfriet, Derivation trees of ground term rewriting systems, Department of Computer Science, Technical Report 96-25, Leiden University, The Netherlands, 1996.
[8] Z. Fülöp, S. Vágvölgyi, Congruential tree languages are the same as recognizable tree languages — a proof for a theorem of D. Kozen, Bull. EATCS 39 (1989) 175–185.
[9] Z. Fülöp, S. Vágvölgyi, Ground term rewriting rules for the word problem of ground term equations, Bull. EATCS 45 (1991) 186–201.
[10] Z. Fülöp, S. Vágvölgyi, Minimal equational representations of recognizable tree languages, Acta Inform. 34 (1997) 59–84.
[11] F. Gécseg, M. Steinby, Tree Automata, Akadémiai Kiadó, Budapest, 1984.
[12] D. Kapur, Shostak's Congruence Closure as Completion, in Rewriting Techniques and Applications: Proc. of the 8th Internat. Conf. Sitges, Spain, 1997, Lecture Notes in Computer Science, vol. 1232, Springer Publishing Company, New York, 1997, pp. 23–37.
[13] D. Kozen, Complexity of finitely presented algebras, Proc. of the 9th Annual ACM Symp. on Theory of Computing, Boulder, Colorado, 1977, pp. 164–177.
[14] M. Oyamaguchi, The Church–Rosser property for ground term rewriting systems is decidable, Theoret. Comput. Sci. 49 (1987) 43–79.
[15] D. Plaisted, A. Sattler-Klein, Proof lengths for equational completion, Inform. and Comput. 125 (1996) 154–170.

[16] I. Rival, B. Sands, A note on the congruence lattice of a finitely generated algebra, Proc. AMS 72 (1987) 451–455.
[17] W. Snyder, A fast algorithm for generating reduced ground rewriting systems from a set of ground equations, J. Symbolic Comput. 15 (1993) 415–450.
[18] S. Vágvölgyi, A fast algorithm for constructing a tree automaton recognizing a congruential tree language, Theoret. Comput. Sci. 115 (1993) 391–399.
[19] S. Vágvölgyi, Simple and Minimal Ground Term Equation Systems, Fundam. Inform. 36 (1998) 367–413.
[20] S. Vágvölgyi, Congruential Complements of Ground Term Rewrite Systems, Theoret. Comput. Sci., to appear.
[21] S. Vágvölgyi, The ground tree transducer game, Fundam. Inform. 34 (1998) 175–201.